Dynamic Expansion of M:N Protection Groups in GMPLS Optical Networks

David W. Griffith and SuKyoung Lee
National Institute of Standards and Technology (NIST)
100 Bureau Drive, Stop 8920
Gaithersburg, MD 20899-8920
{david.griffith,sukyoung}@nist.gov [*]

Abstract

In order to provide reliable connections across metropolitan and wide-area optical networks, the network operator must provide some degree of redundancy so that traffic can be switched from damaged working paths to backup paths that are disjoint from the working paths that they are protecting. In the most general form of path protection, N working paths between two client edge nodes are protected by M backup paths. The set of working and protection paths forms a M:N protection group. In the near future, optical transport networks (OTNs) will use an automated control plane to set up, tear down, or modify connections between client edge nodes. If protection groups are allowed to evolve over time, with working and backup paths being set up or torn down individually, it may be necessary to modify other working and backup paths in addition to those that are being created or destroyed, in order to maximize network utilization. In this paper, we examine the mechanisms that can support adaptive M:N protection group management and describe how existing Generalized Multi-Protocol Label Switching (GMPLS) signaling protocols allow this capability to be deployed in the OTN.

1 Introduction

Optical networks are being redefined by standards bodies such as the International Telecommunications Union (ITU) and the Internet Engineering Task Force (IETF). While SDH/SONET systems will continue to play an important role in both metro and wide-area transport networks, there has been a strong movement, especially over the last three years, toward networks composed of optical switching elements whose control plane incorporates portions of the Internet Protocol (IP) stack. The main driver of this phenomenon is Generalized Multiprotocol Label Switching (GMPLS) [1] and its attendant protocol suite. GMPLS extends the idea of label switching of network layer packets developed in Multiprotocol Label Switching (MPLS) [2] by treating an optical crossconnect's (OXCs') internal mappings of input port/wavelength/subchannel to output port/wavelength/subchannel as analogous to the label mappings that take place within a Label Switch Router (LSR) in a MPLS-capable network. Signaling of new connections in GMPLS networks is accomplished by using either the RSVP-TE [3] or CR-LDP [4] protocols, which consist respectively of extensions to the RSVP protocol [5] that was developed for use with IntServ and DiffServ networks and the Label Distribution Protocol (LDP) [6] that was devised to manage label mappings in networks of LSRs.

In order to make optical transport networks (OTNs) with GMPLS control viable, it is necessary to have a fault recovery regime whose performance is similar to that of the Automatic Protection Switching (APS) recovery mechanism found in SDH/SONET ring networks. Recovery in GMPLS OTNs is complicated by the fact that such networks are anticipated to have more complex physical layer topologies than the linear and ring topologies typically encountered in SDH/SONET networks. For this reason, the recovery architectures being developed in the ITU and IETF prescribe the use of span and end-to-end protection. In both types of protection, additional bandwidth resources in the network are reserved to serve as backup subpaths or paths in the event of a failure on a working path (i.e. a lightpath that is carrying priority traffic). In the most general case of end-to-end protection, a set of N disjoint working paths sharing a common pair of ingress and egress nodes supported by M backup paths form a M:N protection group, where $M \leq N$. In the absence of a failure, the resources associated with the backup paths may be used to carry low priority traffic; if a failure occurs the extra traffic is preempted when the backup

[*] This work was supported, in part, by the National Institute of Standards and Technology (NIST), the Advanced R&E Activity (ARDA), the Laboratory for Telecommunications Sciences (LTS) MENTER project, the Defense Advanced Research Projects Agency (DARPA) Fault Tolerant Networks (FTN) program, and the National Communications System (NCS).

path is seized for use by the traffic that was on the failed working path.

While connections across OTNs are typically of very long duration, on the order of months or years, in certain situations a the network operator may wish to add additional working or protection paths to those already present in an $M{:}N$ protection group. The protection group may be exclusively dedicated to a particular client at the network edge who needs additional capacity or who wishes to enhance the reliability of his connections by having more backup resources made available. In both cases, the network operator will need to provision additional network resources so that the new paths are disjoint from all the previously established paths in the protection group. In some cases, it may not be possible to do this without disrupting the traffic on one or more working paths. However, the traffic management facilities in the GMPLS signaling protocols can be used to support the reallocation of resources to accommodate new working paths while not disturbing the traffic on existing working paths. In this paper, we describe a mechanism for expanding protection groups in a manner that is transparent to the end users. We use features of a routing heuristic that finds multiple edge-disjoint paths to support creation of new working and protection paths. We carry out the signaling to support the protection group modification by using either the RSVP-TE or CR-LDP signaling protocols for GMPLS.

The remainder of this paper is organized as follows. In Section 2 we describe the K-best paths routing heuristic and highlight the features that we will use to carry out sequential working path setup. In Section 3 we describe an algorithm that can be used to incorporate new working paths into an established protection group. Finally in Section 4 we discuss how the GMPLS signaling protocols can be used to support sequential path setup.

2 The K-Best Routing Algorithm

For certain applications, such as establishing protection groups, a network operator needs to create multiple LSPs that share the same endpoints but do not have any other network resources in common. In other cases, LSPs are allowed to share nodes in the network graph but not any edges. This second, less restrictive case, known as "edge-disjointness," is what we consider in this paper. In such a situation, the operator needs to simultaneously route K LSPs while ensuring their disjointness. Suurballe solved this problem in 1974 [7], and a heuristic approach was recently proposed by Bhandari [8]. We provide a brief summary of the workings of Bhandari's algorithm to motivate the development of our algorithm in Section 3.

A network can be represented as a graph $G(\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ and $\mathcal{E}$ are the sets of vertices (or nodes) and edges, respectively. Each bidirectional edge $e \in \mathcal{E}$ represents all the connections between two connected nodes. Thus we assume that all the fiber connections between a given pair of OXCs are routed through a common fiber bundle or conduit, so that the operator would consider routing two working or protection paths in a $M{:}N$ protection group between a given pair of OXCs to be unacceptably risky.

To formalize our discussion, we introduce definitions of routes and route overlaps that were not used in [8]. A route R is a sequence of elements of $\mathcal{N}$, $R = \{n_I, n_1, n_2, \ldots, n_L, n_E\}$. n_I is the ingress node for the route, while n_E is the egress node. There are $L \geq 0$ intermediate nodes on the route. A *subroute* of a route R is a subsequence derived from the sequence of intermediate nodes $\{n_i\}_{i=1}^{L}$ and has the form $\{n_{i+j}\}_{i=1}^{\ell}$ for some $j \geq 0$ and $\ell \leq L - j$.

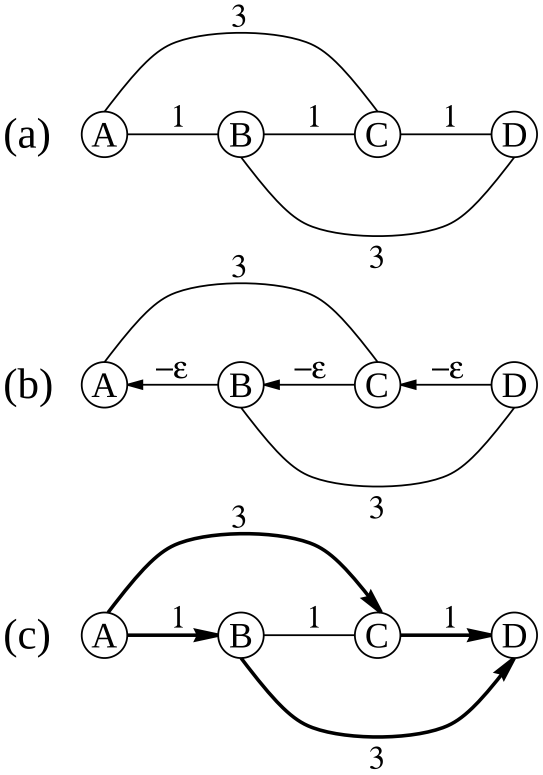

Figure 1. (a): Example network with link costs as shown. (b): The network with the edges associated with the route $R_1 = \{A, B, C, D\}$ removed and the reverse edges assigned a small negative cost. (c): The two paths after computation of route R_2 based on the subgraph in (b) and route splicing.

The heuristic K-best algorithm computes the first of $K \geq 2$ paths by using the Shortest Path First (SPF) algorithm. For example, in the network shown in Fig. 1(a), if a connection between ingress node A and egress node D is needed, the SPF algorithm returns $R_1 = \{A, B, C, D\}$. Once the route for the first path, R_1, has been determined, the directional edges (n_I, n_1), (n_1, n_2) ,. . ., (n_L, n_E), are removed from the graph. The corresponding upstream edges (n_1, n_I), (n_2, n_1), . . ., (n_E, n_L), are assigned a small negative cost $-\varepsilon$. This manipulation of the graph G is shown in Fig. 1(b). The second route R_2 is computed using the modified SPF algorithm given in [8] and the modified network graph. In the case of the network depicted in Fig. 1, this will yield $R_2 = \{A, C, B, D\}$.

The final phase of the Bhandari algorithm modifies the newly computed path if it travels over any edges that have negative weights, because these edges are associated with elements of the set of previously computed routes. Consider the case where we are computing the k^{th} route in a set of K routes. We define an *overlap* to be a subroute $O = \{o_1, o_2, \ldots, o_p\}$ of the new route R_k such that the reverse sequence of nodes $\Omega = \{o_p, o_{p-1}, \ldots, o_2, o_1\}$ is a subroute of an existing route R_i, where $i = 1, 2, \ldots k - 1$. A newly computed route may contain multiple overlaps; it may overlap with an existing route more than once, or it may overlap with multiple existing routes. We define $\mathcal{O} = \{O_j\}_{j=1}^{J}$ to be the set of overlaps contained in route R_k.

The algorithm processes overlaps sequentially, beginning with the one that is most upstream, i.e. closest to the ingress. Let R_a be the previously computed route that overlaps with new route R_k at overlap O, whose length is p. We can partition both routes into sequences of subroutes as follows:

$$\begin{aligned} R_k &= \{R_k^U, O, R_k^D\} \\ R_a &= \{R_a^U, \Omega, R_a^D\}. \end{aligned}$$

An example of this partitioning is shown for the case of two routes R_A and R_B in Fig. 2. R_A was established first, and R_B overlaps R_A at the nodes indicated in the figure. Here R_k^U is the subroute that lies upstream of O and includes n_I, and R_k^D is the subroute that lies downstream of O and includes n_E. R_a^U and R_a^D are defined similarly with respect to Ω. We form two new routes $R_a^{(1)}$ and $R_k^{(1)}$ from R_a and R_k as follows:

$$\begin{aligned} R_k^{(1)} &= \{R_a^U, o_p, R_k^D\} \\ R_a^{(1)} &= \{R_k^U, o_1, R_a^D\}. \end{aligned}$$

We also define $R_i^{(1)} = R_i$ for $i \neq a$. If there are additional overlaps lying downstream of O in R_k, they will be contained in $R_k^{(1)}$. $R_a^{(1)}$ will not overlap with any other elements of $\{R_i^{(1)}\}_{i=1}^{k-1}$, but it may overlap with $R_k^{(1)}$. The next overlap, if one exists, is processed by splicing subroutes of $R_k^{(1)}$ and an element of $\{R_i^{(1)}\}_{i=1}^{k-1}$. The algorithm continues this splicing operation for each overlap in $\mathcal{O}$, which will result in J modifications to R_k and to the previously computed $k-1$ routes. Once all overlaps have been processed, $R_k^{(J)}$ is added to the set of previously computed routes, which becomes $\{R_i^{(J)}\}_{i=1}^{k}$.

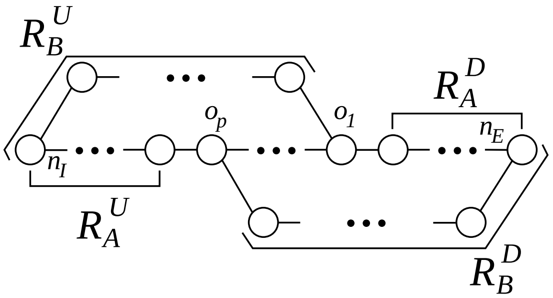

Figure 2. Two routes R_A and R_B with a p-node overlap.

To compute the $(k + 1)^{\text{st}}$ route, the algorithm starts with the original graph G and deletes the downstream-oriented edges associated with each of the k previously-computed routes, and assigns small negative weights to the corresponding upstream-oriented edges. The new route is computed and any overlaps are eliminated. The algorithm terminates when all K routes have been established or when the SPF algorithm is unable to find a path from n_I to n_E in the modified network graph. In the network in Fig. 1, there is just one overlap, $O = \{C, B\}$. Thus $R_1^U = R_2^U = A$ and $R_1^D = R_2^D = D$, and after splicing the new routes are $R_1^{(1)} = \{A, C, D\}$ and $R_2^{(1)} = \{A, B, D\}$. There are no more overlaps, and so the output of the algorithm is $R_1^{(1)}$ and $R_2^{(1)}$. The routes are shown in Fig. 1(c).

When all K routes have been computed in this fashion, they are set up by sending out either Path messages or Label Request messages, depending on whether the network supports RSVP-TE or CR-LDP GMPLS signaling, respectively. Note that in this approach all K routes are established and modified in software before any network resources are committed to support them. We propose to use Bhandari's algorithm to compute new routes and, if necessary, modify existing routes that are already carrying network traffic so that the new routes can be accommodated.

The algorithm described in this section can be extended to the case where multiple edges connect node pairs. This is particularly relevant to optical network architectures, as future optical cross connects are projected to support many fiber connections that may not be routed through a common fiber bundle or conduit and thus may belong to different Shared Risk Link Groups (SRLGs). SRLGs are logical constructs that are used to facilitate diverse routing of optical paths, and their use is described in [11]. In such a case all the directional edges that compose a SRLG between two OXCs are grouped into a single logical bidirectional edge. If the K-best heuristic routes a path R_i over one of the edges in the logical edge, then when the next route R_{i+1} is computed all the downstream edges in the logical edge are deleted from the network graph while all the upstream edges are assigned negative weights, as shown in Fig. 3. In the figure, a route has been computed that passes through OXCs A and B at the fiber ports indicated in the figure. If another route needs to be computed that is edge-disjoint from this route, all fibers entering the ports entered by the route are deleted from the network graph and all fibers exiting those ports are assigned weight $-\varepsilon$. We compute the new route using the modified graph as before and the algorithm then removes overlaps with existing routes in the same manner as before, but we must modify our definition of an overlap, since in this situation it is possible to have two routes traverse the same set of nodes and yet be edge-disjoint. By defining a route to be a sequence of logical edges, $R_k = \{e_1, e_2, \ldots, e_L\}$, we can say that R_k overlaps a previously computed route R_a if R_a contains an edge sequence $\{e_{i+p}, e_{i+p-1}, \ldots, e_{i+1}, e_i\}$ that is the reverse of a subsequence of R_k.

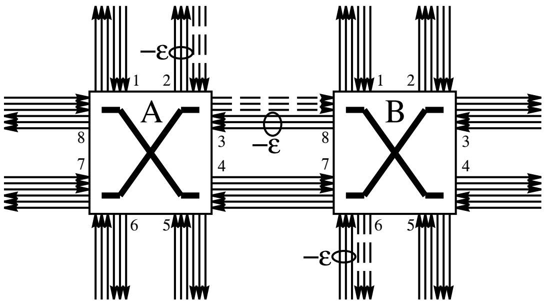

Figure 3. An example of network graph modification after a route passing through OXCs A (entering at port 2 and leaving at port 3) and OXC B (entering at port 8 and leaving at port 6) has been created and added to the set of computed routes by the K-best algorithm. All edges entering port 2 and leaving port 3 of OXC A, and entering port 8 and leaving port 6 of OXC B have been removed even though only one edge in each group is being used for the new path.

3 The Algorithm for Protection Group Expansion

In order to support $M{:}N$ protection group management, the group's ingress node must have knowledge of the routes taken by the existing working and protection paths that compose the group. This information can be obtained by using the Record Route object (RRO) in either RSVP-TE or CR-LDP. The RRO carried back to the ingress node from the egress node (in either a RSVP-TE Resv message or a CR-LDP Label Mapping message) should contain at a minimum the identifiers of the abstract nodes that compose the route.

In the most general case, we wish to add an additional N' paths, so that the protection group will consist of $N + N'$ working paths. This may produce an unacceptable dilution of the protection capability of the group, however. Thus the network operator may wish to create additional protection paths so that the original ratio of working to protection paths is preserved. In such a case the number of additional protection paths that should be created is

$$M' = \left\lceil \frac{MN'}{N} \right\rceil .$$

The additional working and protection paths are computed at the ingress node using the heuristic described in Section 2. The set of existing routes associated with a protection group is $\mathcal{R}(M,N) = \{\{R_{P,i}\}_{i=1}^{M}, \{R_{W,j}\}_{j=1}^{N}\}$, where $R_{P,i}$ is the i^{th} protection paths and $R_{W,j}$ is the j^{th} working path. We will use the K-best algorithm to create a new set of working and protection paths $\mathcal{S}(M,N) = \{\{S_{P,i}\}_{i=1}^{M'}, \{S_{W,j}\}_{j=1}^{N'}\}$ that will be appended to the existing path sets in the protection group. The number of splicing operations due to overlaps needed to create the paths in $\mathcal{S}$ is

$$\mathcal{J} = \sum_{i=1}^{M'} J_{P,i} + \sum_{j=1}^{N'} J_{W,j} \geq 0,$$

where $J_{P,i}$ and $J_{W,j}$ are the number of splicings needed to form the i^{th} protection path and j^{th} working path, respectively. Once the set $\mathcal{S}(M,N)$ is formed, we will have a possibly modified set of working and protection routes $\mathcal{R}^{(\mathcal{J})}(M,N) = \{\{R^{(\mathcal{J})}_{P,i}\}_{i=1}^{M}, \{R^{(\mathcal{J})}_{W,j}\}_{j=1}^{N}\}$.

We now describe the method for modifying existing working and protection paths and adding new ones.

1. For each j, $1 \leq j \leq N$, compare $R_{W,j}$ to $R^{(\mathcal{J})}_{W,j}$. If they are the same, do nothing. If they are different, create a LSP Setup message using the route $R^{(\mathcal{J})}_{W,j}$ and transmit it. The existing route $R_{W,j}$ is not affected at this time.

2. For each path that required a new Setup message, if a Setup Response message is received indicating successful setup, begin transmitting data on the new path, while continuing to transmit data on the existing path. If a failure occurs on the provisional path at this stage, do not transition the traffic to one of the backup paths. If setup fails (e.g. a connection request from another ingress node has seized resources along the provisional route), move on to the next working path and do not delete the existing path.

3. Once the provisional path $R_{W,j}^{(\mathcal{J})}$ is created, tear down the existing path $R_{W,j}$. If a failure occurs on the existing path at this stage, do not transition the traffic to one of the backup paths.

4. For each i, $1 \leq j \leq M$, compare $R_{P,i}$ to $R_{P,i}^{(\mathcal{J})}$. If they are the same, do nothing. If they are different, create a LSP Setup message using the route $R_{P,i}^{(\mathcal{J})}$ and transmit it. The existing route $R_{P,i}$ is not affected at this time. Do not transition traffic from a failed working path to the provisional backup path until a Setup Response message is received. If setup fails, move on to the next protection path and do not delete the existing path.

5. For each path that required a new Setup message, if a Setup Response message is received indicating successful setup, tear down the existing path $R_{P,i}$.

6. If the existing working and protection paths have been modified successfully, create the new working and protection paths using standard path setup signaling. If any existing working paths or protection paths that needed to be modified could not be switched over to new paths, determine which elements of $\mathcal{S}(M', N')$ intersect the unmodifiable paths and delete them from $\mathcal{S}(M', N')$, then set up the remaining elements of $\mathcal{S}(M', N')$ using standard signaling, creating the protection paths before the working paths. Re-run the K-best routing algorithm for the subset of $\mathcal{S}(M', N')$ that could not be established and return to Step 1.

4 Implementing the Algorithm with the GMPLS Signaling Protocols

In this section we describe how the RSVP-TE and CR-LDP signaling protocols can be used to carry out the algorithm described in the previous section. The GMPLS signaling framework, described in [12], allows us to support the M:N protection group modification algorithms without defining additional signaling mechanisms.

RSVP-TE supports “make-before-break” provisioning of LSPs using either Fixed Filter (FF) or Shared Explicit (SE) reservation styles, as explained in [3]. SE is preferable because it allows the creation of non-disjoint LSPs that share resources, whereas FF requires the network to allocate additional resources for the new LSP. To set up a provisional optical connection to replace an existing one (say, $R_{W,j}$) that must be modified, the ingress node needs to transmit a Path message containing a LABEL REQUEST Object and an Explicit Route Object (ERO) that contains the addresses of the nodes that compose $R_{W,j}^{(\mathcal{J})}$. As described in the RFC, the ingress node must form a new SENDER_TEMPLATE object using a LSP ID that is different from the one associated with the existing path. The SESSION object will be the same. In this way the ingress appears to be two different sources.

If the provisional alternate route for an existing LSP is successfully created, then the ingress node will initiate teardown of the legacy route by installing an Admin Status Object in a Path message that it sends to the egress node. The egress will reply with a Resv message that contains a similar Admin Status object. The ingress will transmit a PathTear message once it receives the Resv message. This procedure is described in more detail in [13].

In a CR-LDP network, a similar make-before-break operation needs to be carried out. This is done using the mechanisms described in [4]. In this case the sender must transmit a Label Request message along the path for the provisional LSP; the message must carry a LSPID TLV which uses the ingress node address as the source node address but which carries a LSP ID that is different from that associated with the existing LSP. The Action Indicator flags should be set to all zeros since creating the provisional LSP is a new LSP setup. The Label Request must carry an Explicit Route TLV that contains the route $R_{W,j}^{(\mathcal{J})}$. If the provisional LSP is properly established, then the existing LSP is torn down using the deletion procedure described in [14]. The ingress transmits a Notification message that contains an Admin Status TLV with the Reflect and Delete bits set high; the egress will respond with a Label Withdraw message that propagates back upstream to the ingress.

5 Summary and Future Work

In this paper, we have proposed a mechanism by which additional working or protection paths can be created for M:N protection groups in GMPLS optical networks. Using either RSVP-TE or CR-LDP with some of the extensions that are being proposed in the IETF to support optical restoration, this approach will allow the network operator to adjust existing working paths without disturbing the traffic they are carrying. The amount of effort that must be expended to adjust backup baths depends on the degree of soft reservation of resources that exists on such paths. In addition, while the relative infrequency of path adjustments in the network core should make failure of the path adjustment procedure in Section 3 unlikely, we need to do further analysis and simulate the behavior of the algorithm to determine how a given probability of path adjustment failure impacts the time required to complete the $M{:}N$ protection group expansion process.

References

[1] E. Mannie (ed.). Generalized multi-protocol label switching (GMPLS) architecture. IETF Internet Draft, March 2002.

[2] E. Rosen, A. Viswanathan, and R. Callon. Multiprotocol label switching architecture. IETF RFC 3031, January 2001.

[3] D. Awduce et al. RSVP-TE: Extensions to RSVP for LSP tunnels. IETF RFC 3209, December 2001.

[4] B. Jamoussi (ed.), L. Andersson, et al. Constraint-based LSP setup using LDP. IETF RFC 3212, January 2002.

[5] R. Braden (ed.), L. Zhang, S. Berson, S. Herzog, and S. Jamin. Resource ReSerVation Protocol (RSVP) – Version 1 functional specification. IETF RFC 2205, September 1997.

[6] L. Andersson, P. Doolan, N. Feldman, A. Fredette, and B. Thomas. LDP specification. IETF RFC 3036, January 2001.

[7] J. Suurballe. Disjoint paths in network. *Networks*, 4:125–145, 1974.

[8] R. Bhandari. Optimal physical diversity algorithms and survivable networks. *Proceedings of the Second IEEE Symposium on Computers and Communications, 1997*, Alexandria, Egypt, 1-3 July, 1997, pp. 433-441.

[9] G. Li et al. RSVP-TE extensions for shared-mesh restoration in transport networks. IETF Internet Draft, November 2001.

[10] D. Guo et al. Extensions to RSVP-TE for supporting diverse path protection. IETF Internet Draft, July 2001.

[11] D. Papadimitriou (ed.), F. Poppe, J. Jones, et al. Inference of shared risk link groups. IETF Internet Draft, November 2001.

[12] P. Ashwood-Smith et al. Generalized MPLS – Signaling functional description. IETF Internet Draft, November 2001.

[13] P. Ashwood-Smith et al. Generalized MPLS signaling – RSVP-TE extensions. IETF Internet Draft, November 2001.

[14] P. Ashwood-Smith et al. Generalized MPLS signaling – CR-LDP extensions. IETF Internet Draft, November 2001.